

ANT - DEPLOYING APPLICATIONS

http://www.tutorialspoint.com/ant/ant_deploying_applications.htm

Copyright © tutorialspoint.com

In the previous chapter, we have learnt how to package an application and deploy it to a folder. In this example, we are going to take it even further.

We are going to deploy the web application directly to the application's server deploy folder, then we are going to add few Ant targets to start and stop the services. Let us continue with the **Hello World** fax web application. This is a continuation of the previous chapter, the new components are highlighted in red

build.properties

```
# Ant properties for building the springapp

appserver.home=c:\\install\\apache-tomcat-7.0.19
# for Tomcat 5 use ${appserver.home}/server/lib
# for Tomcat 6 use ${appserver.home}/lib
appserver.lib=${appserver.home}/lib

deploy.path=${appserver.home}/webapps

tomcat.manager.url=http://www.tutorialspoint.com:8080/manager
tomcat.manager.username=tutorialspoint
tomcat.manager.password=secret
```

build.xml

```
<?xml version="1.0"?>

<project name="fax" basedir="." default="usage">
  <property file="build.properties"/>
  <property name="src.dir" value="src"/>
  <property name="web.dir" value="war"/>
  <property name="javadoc.dir" value="doc"/>
  <property name="build.dir" value="${web.dir}/WEB-INF/classes"/>
  <property name="name" value="fax"/>

  <path >
    <fileset dir="${web.dir}/WEB-INF/lib">
      <include name="*.jar"/>
    </fileset>
    <pathelement path="${build.dir}"/>
  </path>

  <target name="javadoc">
    <javadoc packageNames="faxapp.*" sourcepath="${src.dir}"
      destDir="doc" version="true" windowTitle="Fax Application">
      <doctitle><![CDATA[<h1>= Fax Application
        =</h1>]]></doctitle>
      <bottom><![CDATA[Copyright © 2011. All
        Rights Reserved.]]></bottom>
      <group title="util packages" packages="faxapp.util.*"/>
      <group title="web packages" packages="faxapp.web.*"/>
      <group title="data packages"
        packages="faxapp.entity.*:faxapp.dao.*"/>
    </javadoc>
  </target>

  <target name="usage">
    <echo message=""/>
    <echo message="${name} build file"/>
    <echo message="-----"/>
    <echo message=""/>
    <echo message="Available targets are:"/>
  </target>
</project>
```

```

    <echo message=""/>
    <echo message="deploy    --> Deploy application
      as directory"/>
    <echo message="deploywar --> Deploy application
      as a WAR file"/>
    <echo message=""/>
  </target>

  <target name="build" description="Compile main
    source tree java files">
    <mkdir dir="${build.dir}"/>
    <javac destdir="${build.dir}" source="1.5"
      target="1.5" debug="true"
      deprecation="false" optimize="false" failonerror="true">
      <src path="${src.dir}"/>
      <classpath ref/>
    </javac>
  </target>

  <target name="deploy" depends="build"
    description="Deploy application">
    <copy todir="${deploy.path}/${name}"
      preservelastmodified="true">
      <fileset dir="${web.dir}">
        <include name="**/*.*/"/>
      </fileset>
    </copy>
  </target>

  <target name="deploywar" depends="build"
    description="Deploy application as a WAR file">
    <war destfile="${name}.war"
      webxml="${web.dir}/WEB-INF/web.xml">
      <fileset dir="${web.dir}">
        <include name="**/*.*/"/>
      </fileset>
    </war>
    <copy todir="${deploy.path}" preservelastmodified="true">
      <fileset dir=".">
        <include name="*.war"/>
      </fileset>
    </copy>
  </target>

  <target name="clean" description="Clean output directories">
    <delete>
      <fileset dir="${build.dir}">
        <include name="**/*.class"/>
      </fileset>
    </delete>
  </target>

```

```

<!-- ===== -->
<!-- Tomcat tasks -->
<!-- ===== -->

<path >
  <!-- We need the Catalina jars for Tomcat -->
  <!-- * for other app servers - check the docs -->
  <fileset dir="${appserver.lib}">
    <include name="catalina-ant.jar"/>
  </fileset>
</path>

<taskdef name="install"
  classname="org.apache.catalina.ant.InstallTask">
  <classpath ref/>
</taskdef>
<taskdef name="reload"
  classname="org.apache.catalina.ant.ReloadTask">
  <classpath ref/>

```

```

</taskdef>
<taskdef name="list"
  classname="org.apache.catalina.ant.ListTask">
  <classpath ref/>
</taskdef>
<taskdef name="start"
  classname="org.apache.catalina.ant.StartTask">
  <classpath ref/>
</taskdef>
<taskdef name="stop"
  classname="org.apache.catalina.ant.StopTask">
  <classpath ref/>
</taskdef>

<target name="reload" description="Reload application in Tomcat">
  <reload url="${tomcat.manager.url}"
    username="${tomcat.manager.username}"
    password="${tomcat.manager.password}"
    path="/${name}"/>
</target>
</project>

```

In this exercise, we have used Tomcat as our application server. First, in the build properties file, we have defined some additional properties.

- The **appserver.home** points to the installation path to the Tomcat application server.
- The **appserver.lib** points to the library files in the Tomcat installation folder.
- The **deploy.path** variable now points to the webapp folder in Tomcat.

Applications in Tomcat can be stopped and started using the Tomcat manager application. The URL for the manager application, username and password are also specified in the build.properties file. Next we declare a new CLASSPATH that contains the **catalina-ant.jar**. This jar file is required to execute Tomcat tasks through Apache Ant.

The catalina-ant.jar provides the following tasks:

Properties	Description
InstallTask	Installs a web application. Class Name: org.apache.catalina.ant.InstallTask
ReloadTask	Reload a web application. Class Name: org.apache.catalina.ant.ReloadTask
ListTask	Lists all web applications. Class Name: org.apache.catalina.ant.ListTask
StartTask	Starts a web application. Class Name: org.apache.catalina.ant.StartTask
StopTask	Stops a web application. Class Name: org.apache.catalina.ant.StopTask
ReloadTask	Reloads a web application without stopping. Class Name: org.apache.catalina.ant.ReloadTask

The reload task requires the following additional parameters. - 1) URL to the manager application 2) Username to restart the web application 3) Password to restart the web application 4) Name of the web application to be restarted

Let us issue the **deploy-war** command to copy the webapp to the Tomcat webapps folder and then let us reload the Fax Web application. The following outcome is the result of running the ant file:

```

C:\>ant deploy-war
Buildfile: C:\build.xml

```

```
BUILD SUCCESSFUL
Total time: 6.3 seconds

C:\>ant reload
Buildfile: C:\build.xml

BUILD SUCCESSFUL
Total time: 3.1 seconds
```

Once the above task is run, the web application is deployed and the web application is reloaded.